C212/A592 Lab 4

Intro to Software Systems

Objectives:

- Using Conditional Statements
- Using loops
- Random Number generator
- GUI Applications
- Junit Testing

Instructions:

- Please review the requirements given below carefully and complete your work. You should compress all source files (including main to test your work) into a zip file and submit it through Canvas. The grading scheme is provided on Canvas.
- You are free to use any looping construct of your choice
- Make sure that you display appropriate messages that are intuitive and clear
- Use Java's random number generator method (limit it to integers only)
- Do NOT write the entire logic of the program inside main(). Create separate methods that perform all the work and then invoke it from main()
- Write Junit Tests for **Exercise 1 and 4** only.
- **Note**: Use of Arrays is not allowed!

Lab Requirements

1. Roman numbers: Write a program that converts a number from Roman number system to positive integer. The Roman number system has digits

I	1
V	5
X	10
L	50
C	100
D	500
M	1,000

   Numbers are formed according to the following rules:

   a) Only numbers up to 3,999 are represented.
   b) As in the decimal system, the thousands, hundreds, tens, and ones are expressed separately.
   c) The numbers 1 to 9 are expressed as

I	1	IV	4	VII	7
II	2	V	5	VIII	8
III	3	VI	6	IX	9

As you can see, an I preceding a V or X is subtracted from the value, and you can never have more than three I's in a row.

d) Tens and hundreds are done the same way, except that the letters X, L, C and C, D, M are used instead of I, V, X, respectively.

Your program should include a java method (with appropriate method name, return type and argument lists) and accompanying test program that works in the following manner: Given an input, such as MCMLXXVIII, and it returns an answer after converting it to decimal 1978.

2. Create a menu-based program to generate the given patterns. A menu-based program is a program that provides the user a Menu and user has an option to select the item from the menu. The program will not stop till user chooses to exit the program with option E. Program should only accept A, B, C and E as valid options. User will also provide a number that will be used as size to draw the chosen pattern. Patterns with size 5 are shown below.

A.

```
1 2 3 4 5
 2 3 4 5
  3 4 5
   4 5
    5
   4 5
  3 4 5
 2 3 4 5
1 2 3 4 5
```

B.

```
    *
   * *
  *   *
 *     *
*********
```

C.

```
    *
   * *
  *   *
 *     *
*       *
 *     *
  *   *
   * *
    *
```

Example Menu:

A. Pattern 1
B. Pattern 2
C. Pattern 3
E. Exit

Note: You MUST NOT hardcode the patterns and you have to use loops (or nested loops) to print the patterns of given size. Be sure to print the empty spaces appropriately to make the pattern look like the ones shown above.

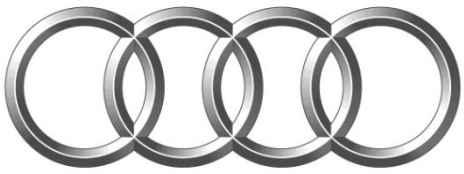
3. Write a java program to generate 10 random numbers between 1 and 10. As you generate each number, keep appending it to a string with a comma after each number. In the end give the user a count of each random number in the string and remove the random number with the highest frequency from your string.
   Note: If two or more numbers have the same frequency then just delete the smallest number. For example if number '2' and number '3' both have frequency of 2 then delete the number '2' from the string.

   Random Numbers = 1,1,2,4,5,6,6,2,1,2
   Frequency:
   1 -> 2
   2 -> 3
   3 -> 1
   5 -> 1
   6 -> 2
   Deleting the Numbers with Max Frequency..
   Random Numbers = 1,1,4,5,6,6,1

4. Write a java function (with appropriate method name, return type and argument lists) and accompanying test program to check if the third string is a valid shuffle of the first two strings. This is true when the third string has the following constraints:
   - The length of 3rd string is equal to length of string1 + length of string2.
   - 3rd string should consist of letters only from string 1 and string 2.

   Sample Output:

   Enter the first String: abc
   Enter the second String: Def
   Enter the third String: dabecf
   Answer: True



5. Write a graphical application that draws Audi Logo rings with colors. A sample is shown above. Be creative with colors.